SECS1024 - Laboratoire 10 - Android Diva

laboratoire not´e 8 points - 10% de la note finale

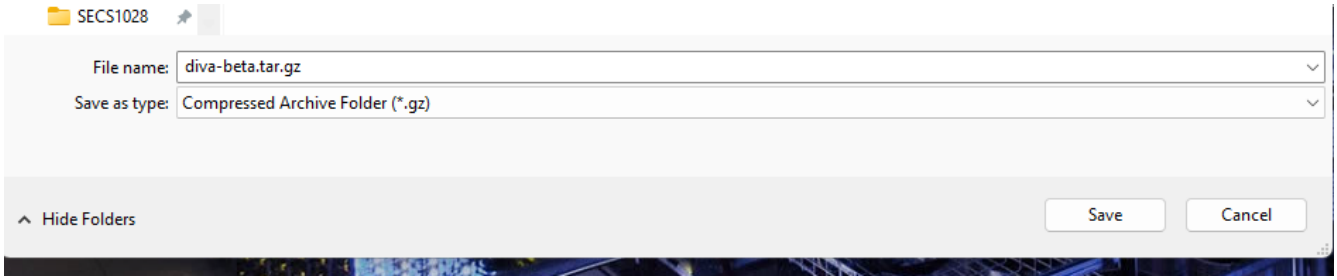
a` rendre pour le 1er avril

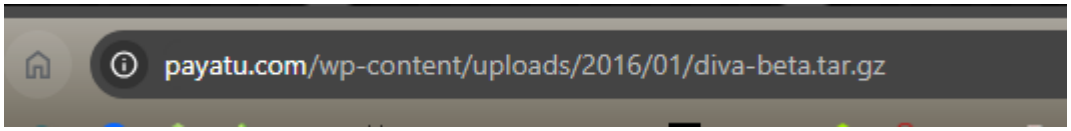
Objectif : trouver des failles de s´ecurit´e sur une application Android. Utilisez pour cela une analyse statique (le code source de l'application est visible sur https://github.com/payatu/divaandroid) et une analyse dynamique sur Android-x86 (en r´eseau interne de VirtualBox).

Mise en place du laboratoire : utilisez pour ce laboratoire une VM Android-x86 connect´ee r´eseau en interne sur VirtualBox.

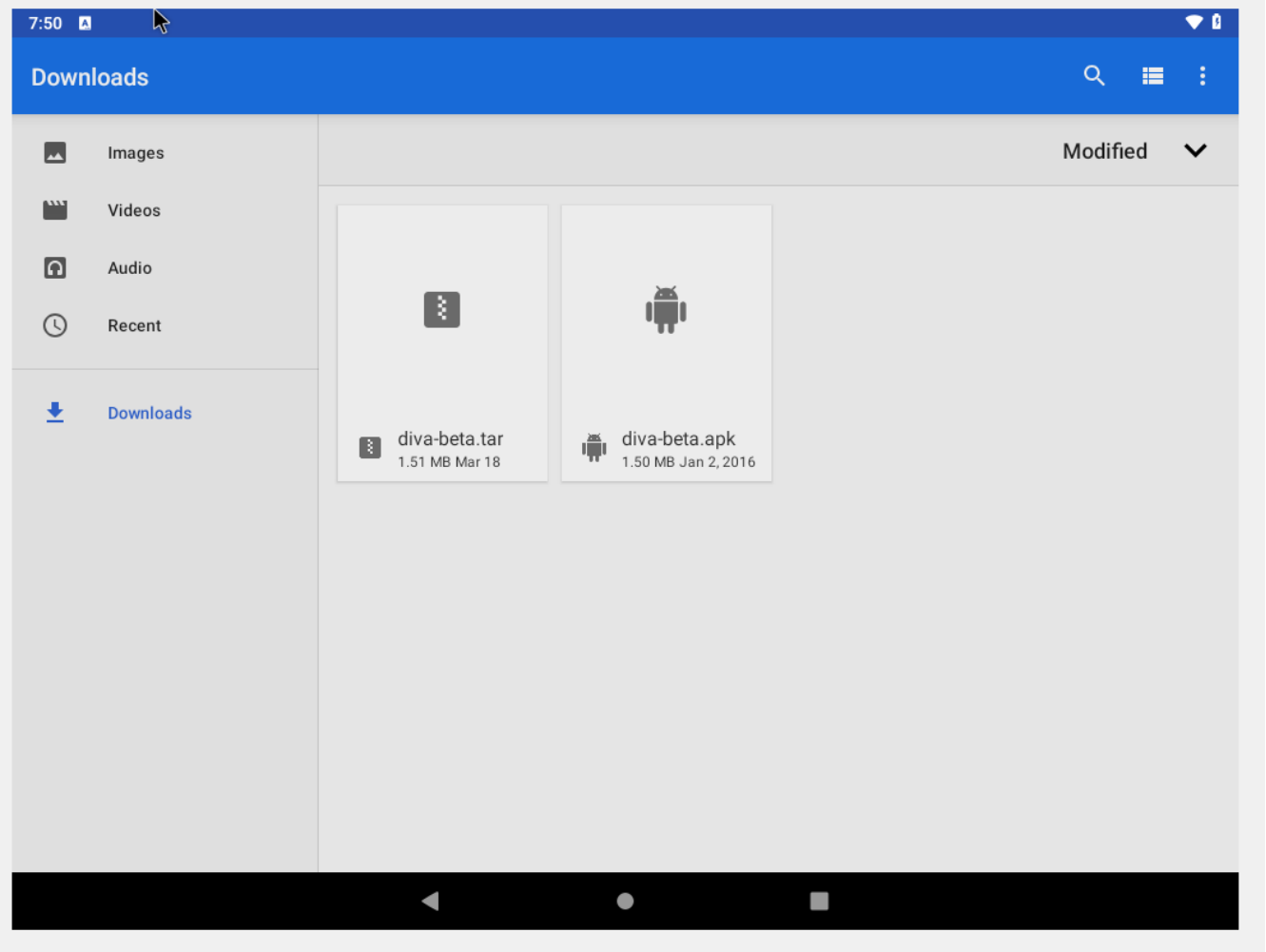
1 Installation de DIVA

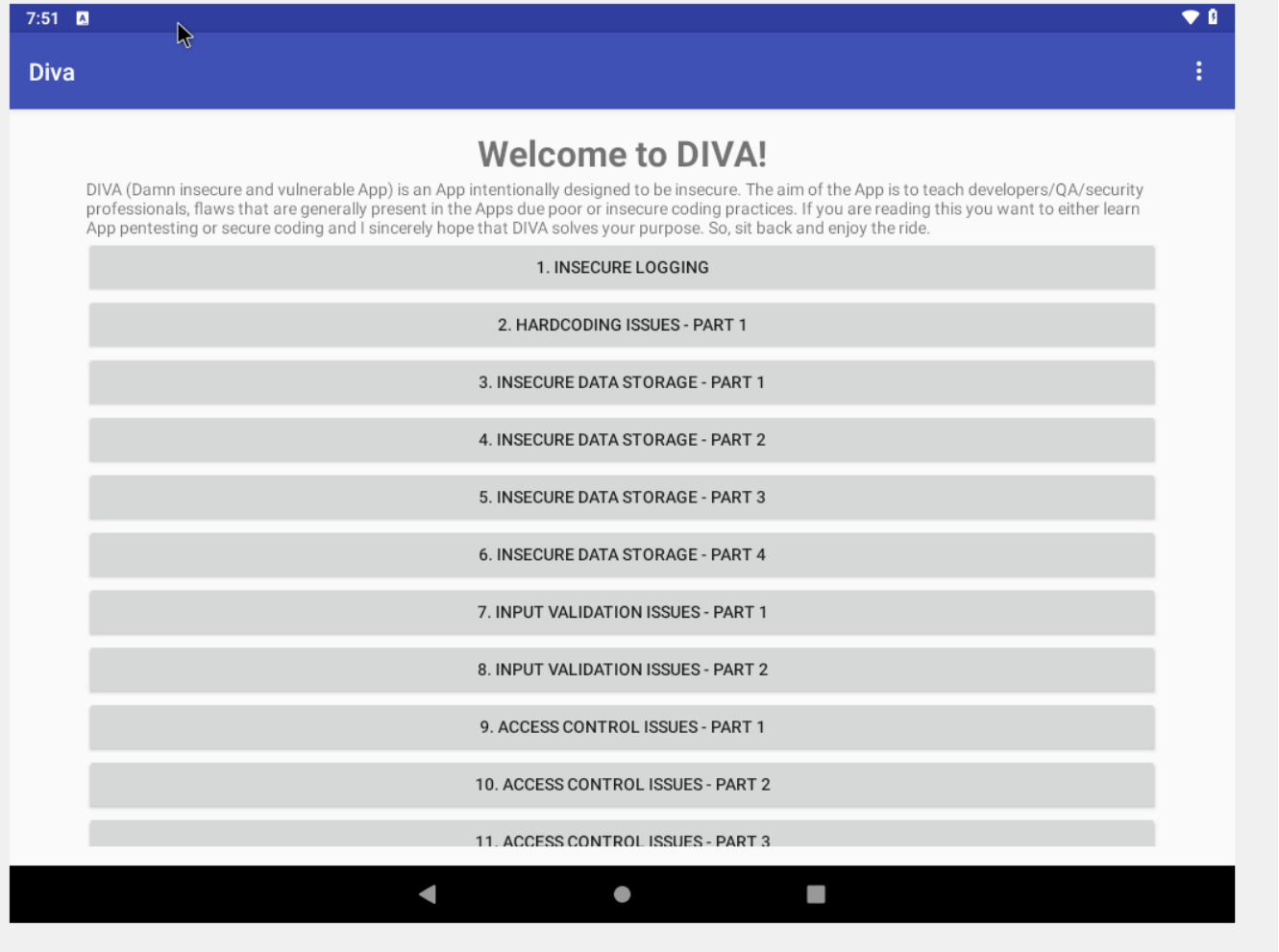
(1 point) Diva est une application Android volontairement vuln´erable. Elle est t´el´echargeabvle sur: http://www.payatu.com/wp-content/uploads/2016/01/diva-beta.tar.gz





1) installez cette application sur Android-x86 (en mode NAT) et faites une capture ´ecran de l'application ouverte. (1 point)

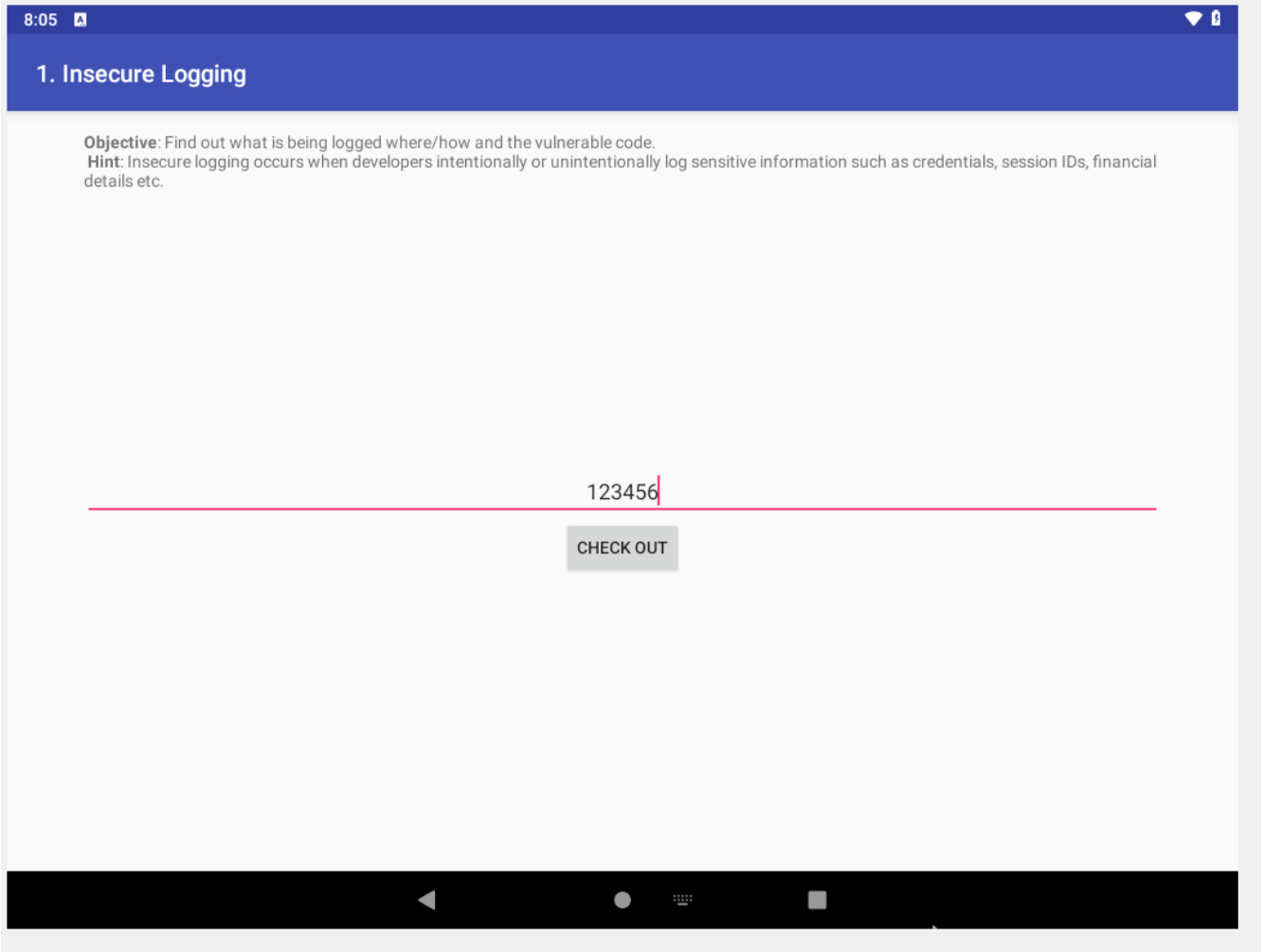




2 Probl`emes de s´ecurit´e (7 points)

Connectez l'application Diva sur le r´eseau interne VB (hors Internet).
Recherchez 7 failles de s´ecurit´e parmi les 13 probl`emes/menus propos´es.
Documentez vos r´eponses par des explications textuelles et des captures ´ecran. (1 point par probl`eme).

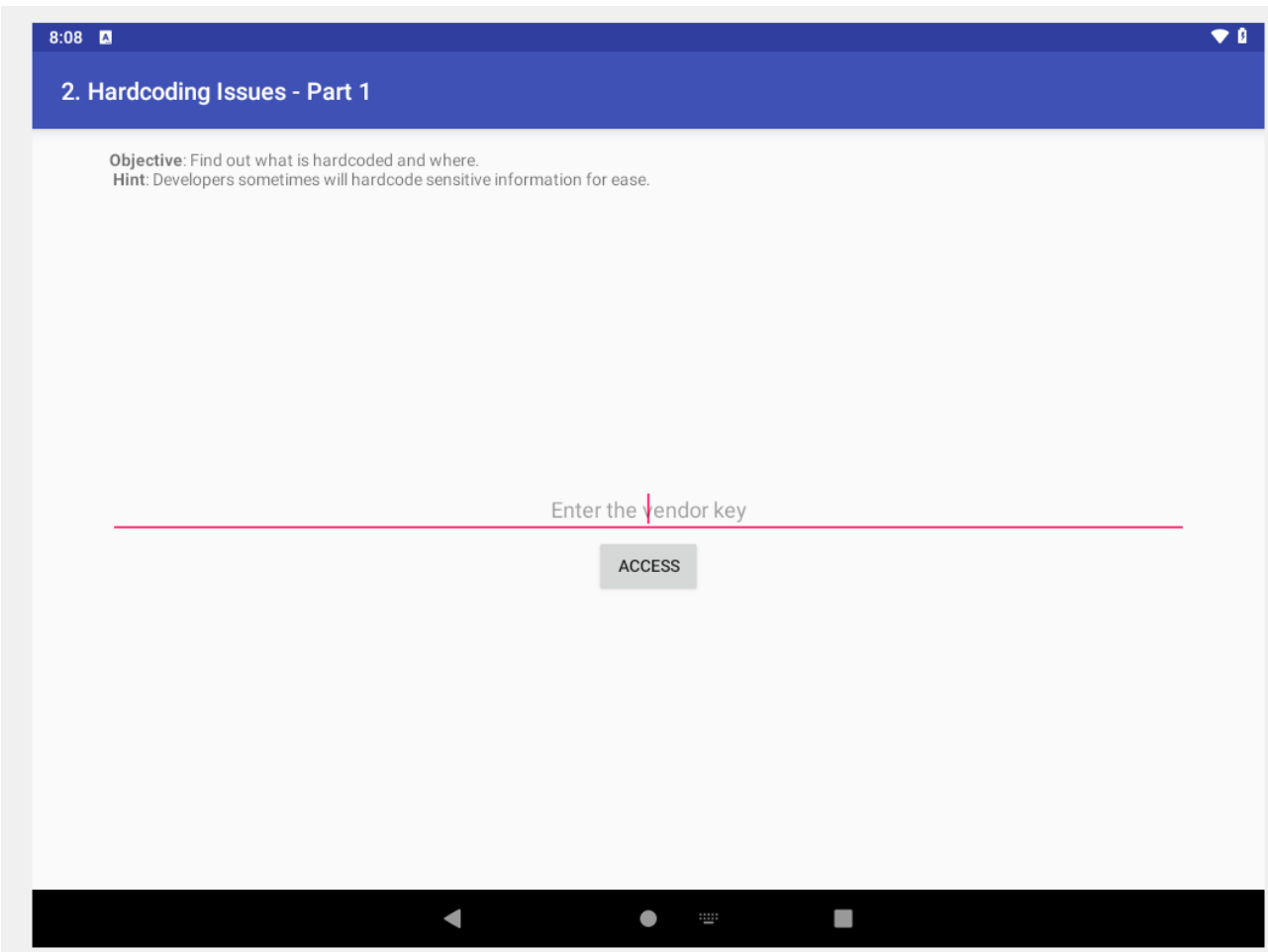
#1 insecure logging

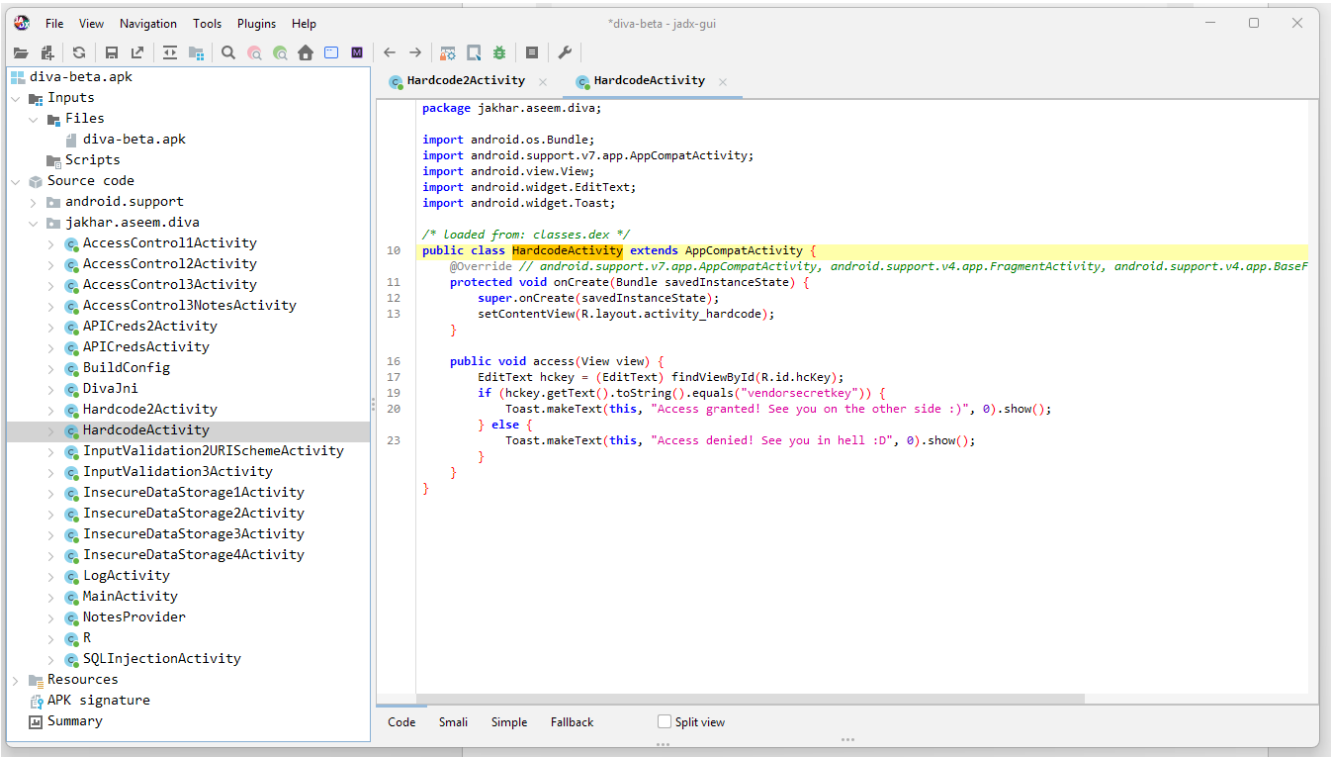


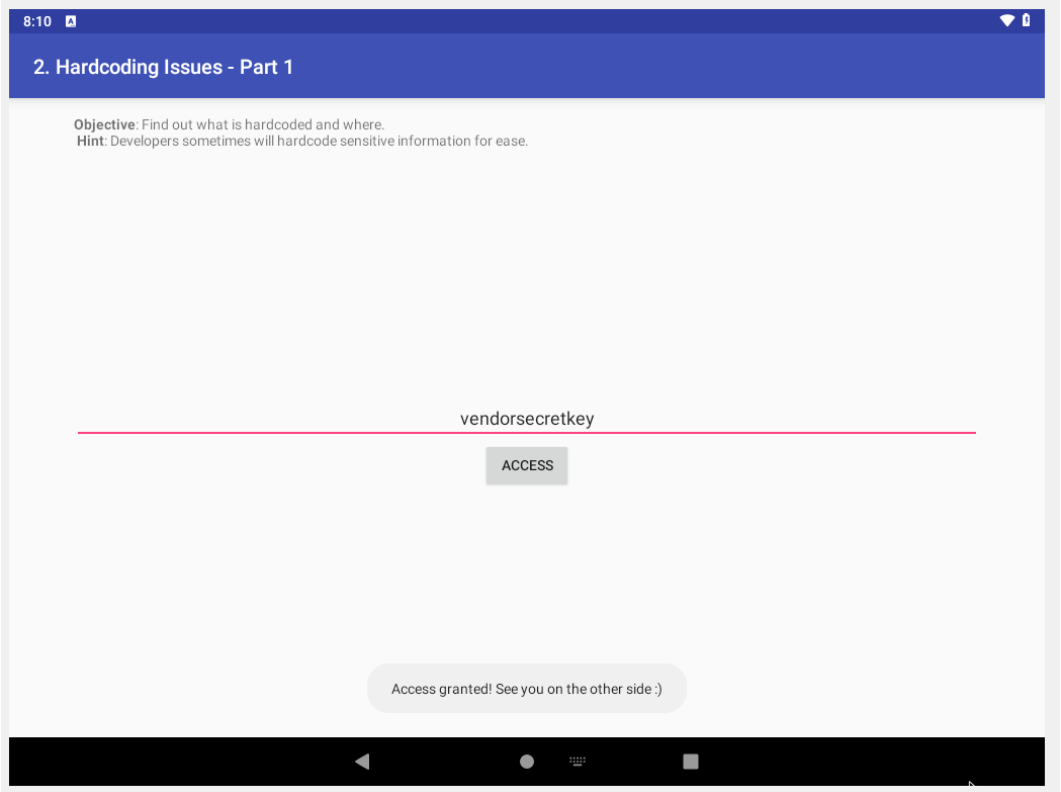

```
  Trusted GRUB now booting 'Android-x86 9.0-r2'

  Progress:  ▒▒▒▒Detecting Android-x86... found at /dev/sda1
console:/ # logcat | grep 123456
^C
130|console:/ # logcat | grep 123456
03-25 08:05:48.173  7321  7321 E diva-log: Error while processing transaction with credit card: 123456
03-25 08:05:49.122  7321  7321 E diva-log: Error while processing transaction with credit card: 123456
```

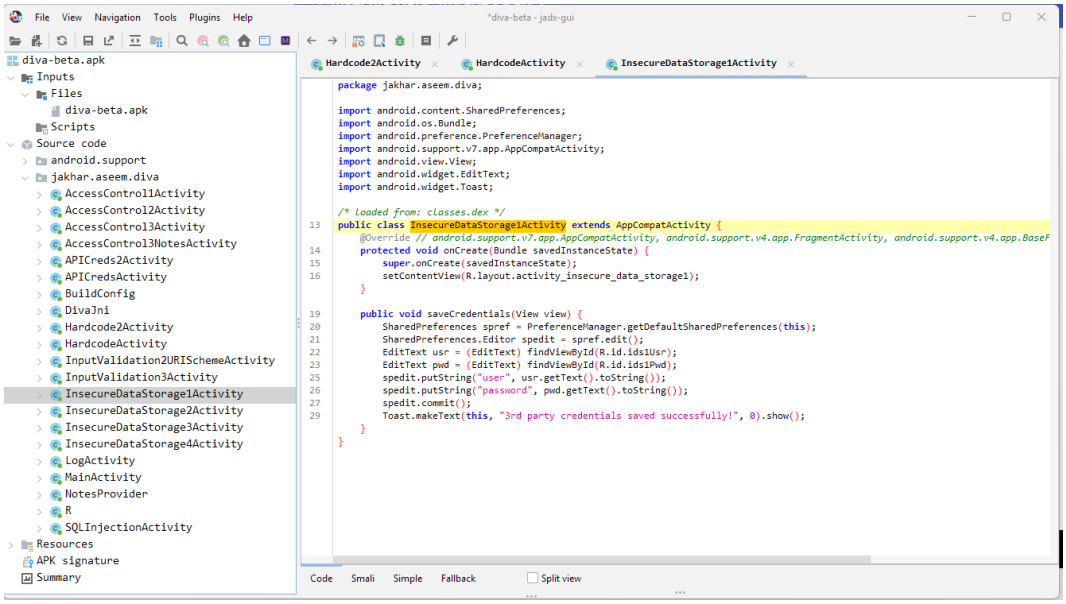
#2 hardcoding issues

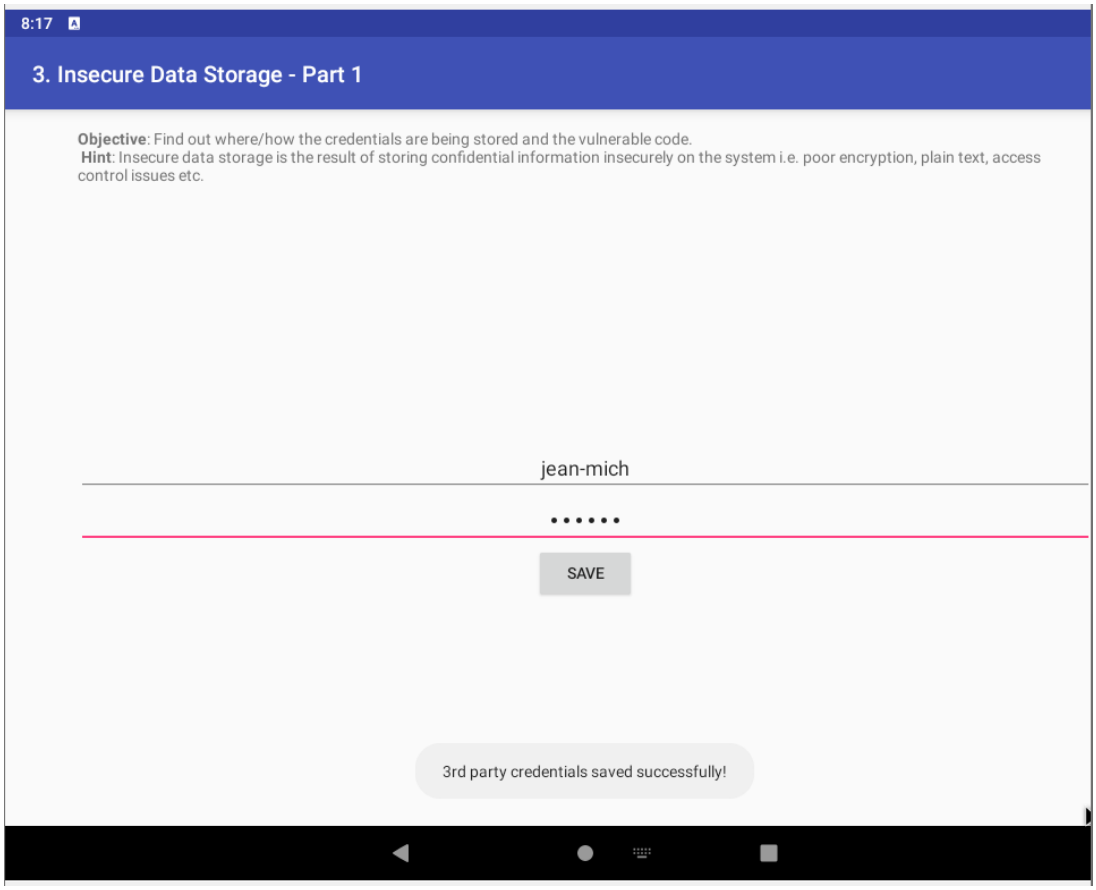

8:08
2. Hardcoding Issues - Part 1
Objective: Find out what is hardcoded and where.
Hint: Developers sometimes will hardcode sensitive information for ease.
Enter the vendor key
ACCESS


File View Navigation Tools Plugins Help
*diva-beta - jadx-gui
diva-beta.apk
Inputs
Files
diva-beta.apk
Scripts
Source code
android.support
jakhar.aseem.diva
AccessControl1Activity
AccessControl2Activity
AccessControl3Activity
AccessControl3NotesActivity
APICreds2Activity
APICredsActivity
BuildConfig
DivaJni
Hardcode2Activity
HardcodeActivity
InputValidation2URISchemeActivity
InputValidation3Activity
InsecureDataStorage1Activity
InsecureDataStorage2Activity
InsecureDataStorage3Activity
InsecureDataStorage4Activity
LogActivity
MainActivity
NotesProvider
R
SQLInjectionActivity
Resources
APK signature
Summary
Hardcode2Activity
HardcodeActivity
package jakhar.aseem.diva;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
/* loaded from: classes.dex */
public class HardcodeActivity extends AppCompatActivity {
@Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.BaseF
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_hardcode);
}
public void access(View view) {
EditText hckey = (EditText) findViewById(R.id.hcKey);
if (hckey.getText().toString().equals("vendorsecretkey")) {
Toast.makeText(this, "Access granted! See you on the other side :)", 0).show();
} else {
Toast.makeText(this, "Access denied! See you in hell :D", 0).show();
}
}
}
Code
Smali
Simple
Fallback
Split view


8:10
2. Hardcoding Issues - Part 1
Objective: Find out what is hardcoded and where.
Hint: Developers sometimes will hardcode sensitive information for ease.
vendorsecretkey
ACCESS
Access granted! See you on the other side :)

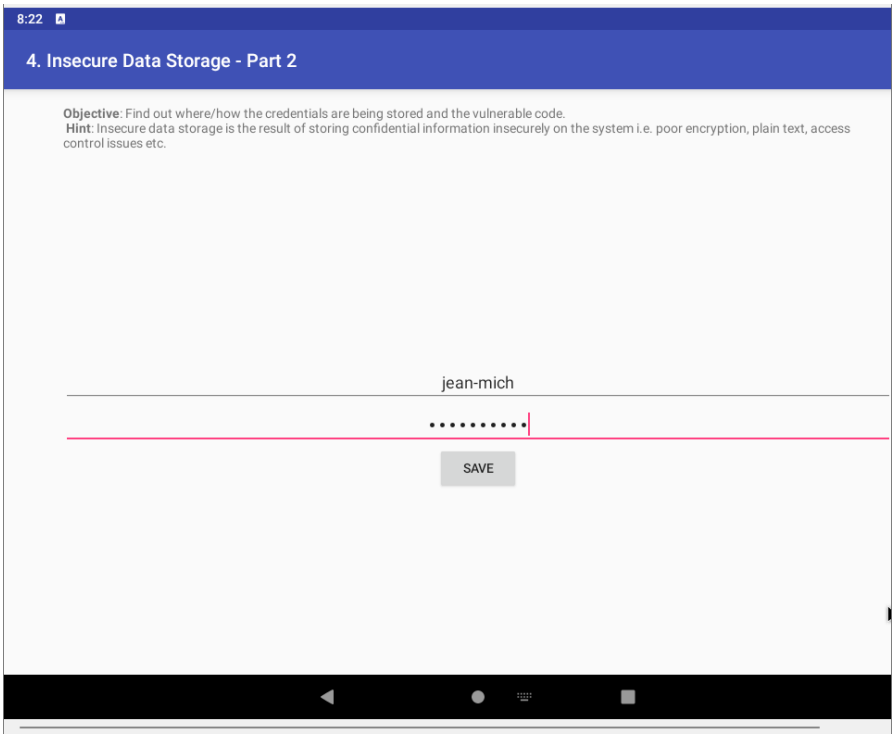
#3 insecure data storage part 1

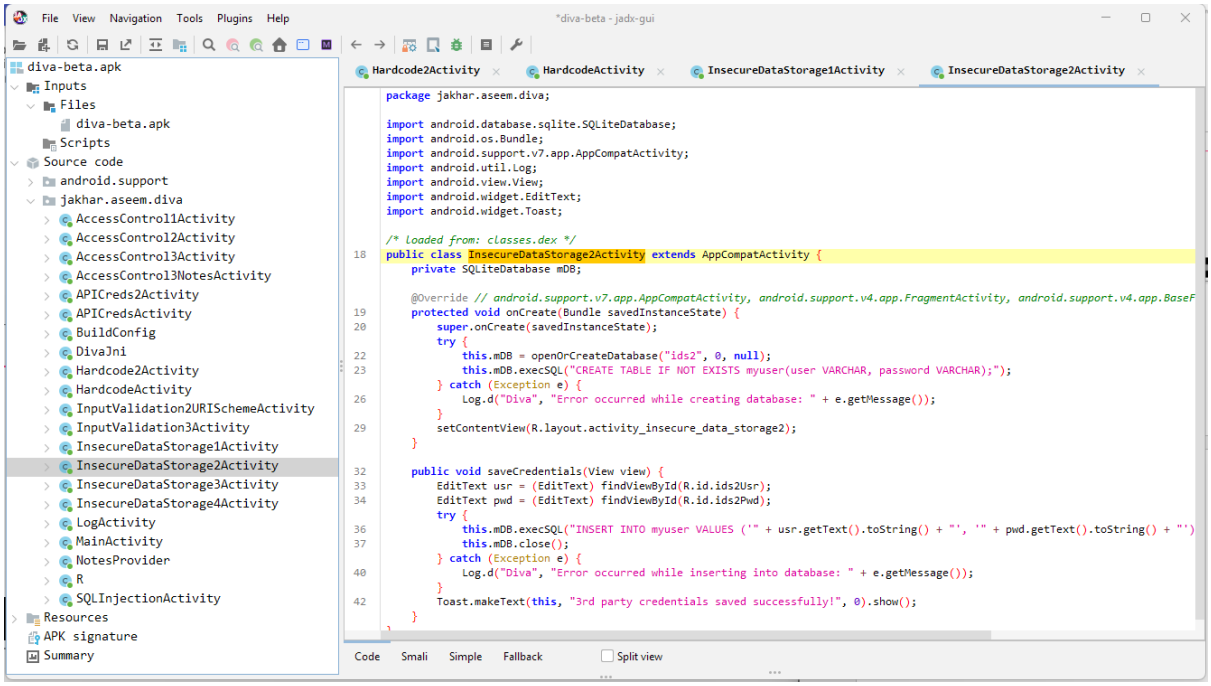




```
console:/data/data # cd /
console:/ # cd data
console:/data # cd data
console:/data/data # cd jakhar.aseem.diva/
console:/data/data/jakhar.aseem.diva # ls
cache code_cache databases
console:/data/data/jakhar.aseem.diva # ls
cache code_cache databases shared_prefs
console:/data/data/jakhar.aseem.diva # cd shared_prefs/
console:/data/data/jakhar.aseem.diva/shared_prefs # ls
jakhar.aseem.diva_preferences.xml
console:/data/data/jakhar.aseem.diva/shared_prefs # cat jakhar.aseem.diva_preferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="password">pierre</string>
    <string name="user">jean-mich</string>
</map>
console:/data/data/jakhar.aseem.diva/shared_prefs # _
```

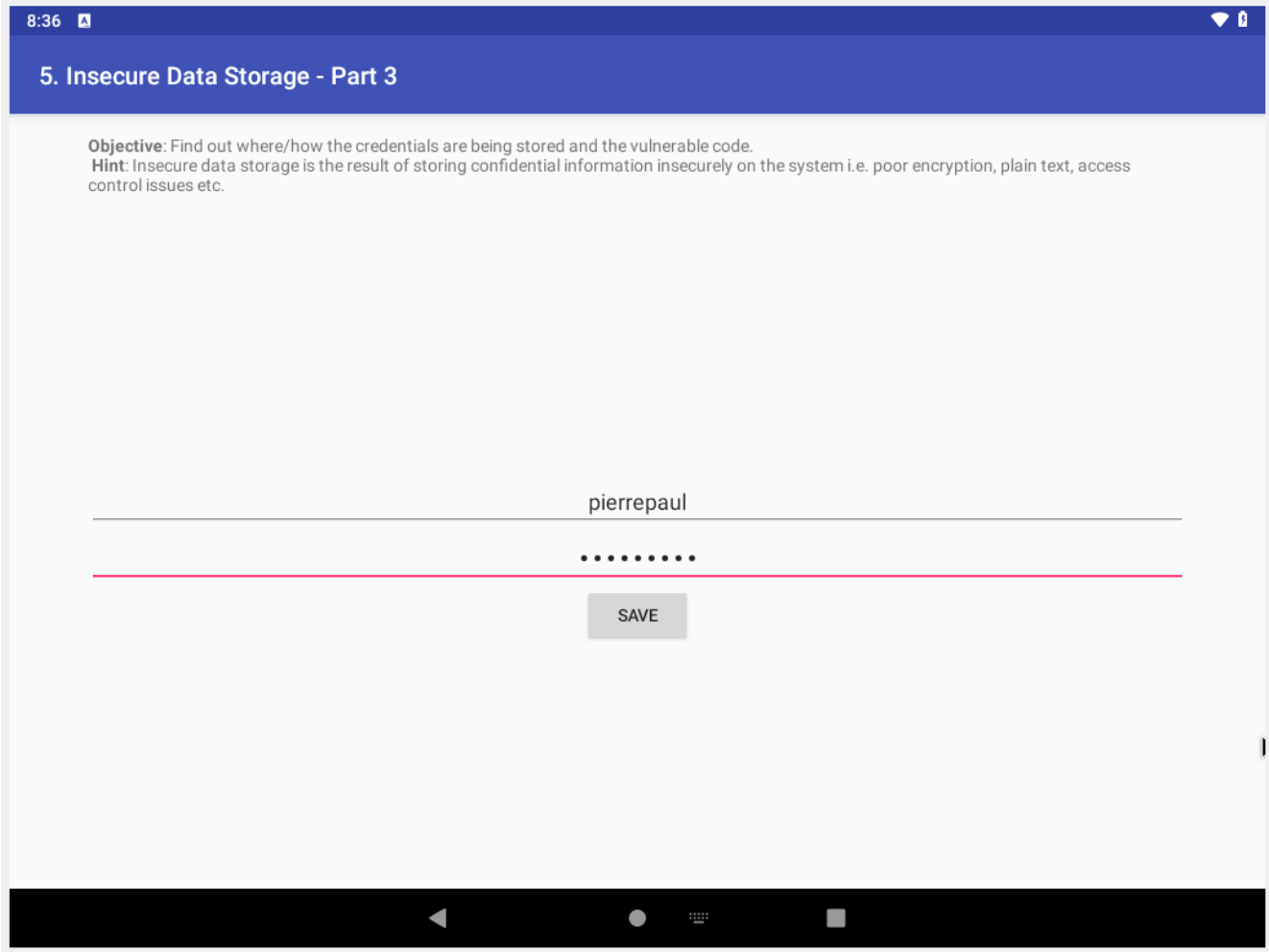
#4 insecure data storage part 2

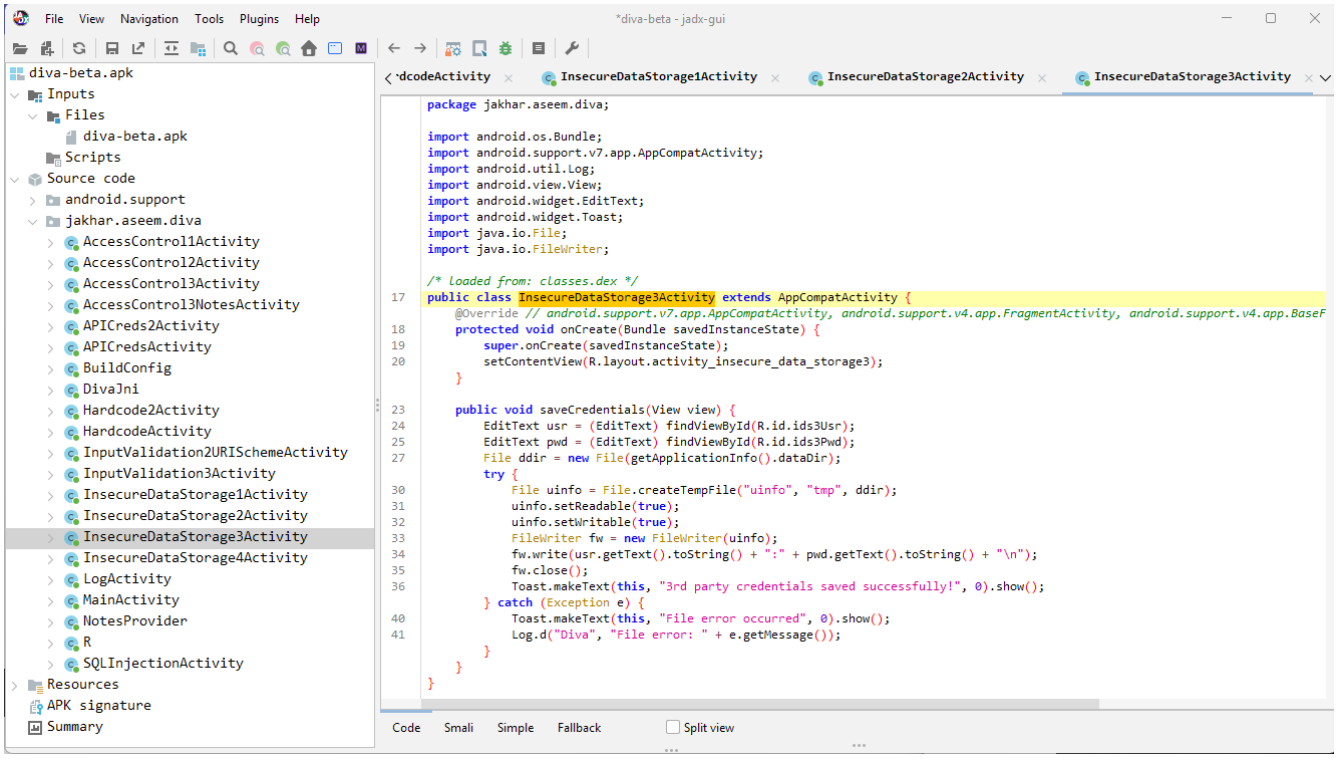





```
console:/data/data/jakhar.aseem.diva/databases # cd ../
console:/data/data/jakhar.aseem.diva # cd databases/
console:/data/data/jakhar.aseem.diva/databases # ls
divanotes.db divanotes.db-shm divanotes.db-wal ids2
console:/data/data/jakhar.aseem.diva/databases # vi ids2
```

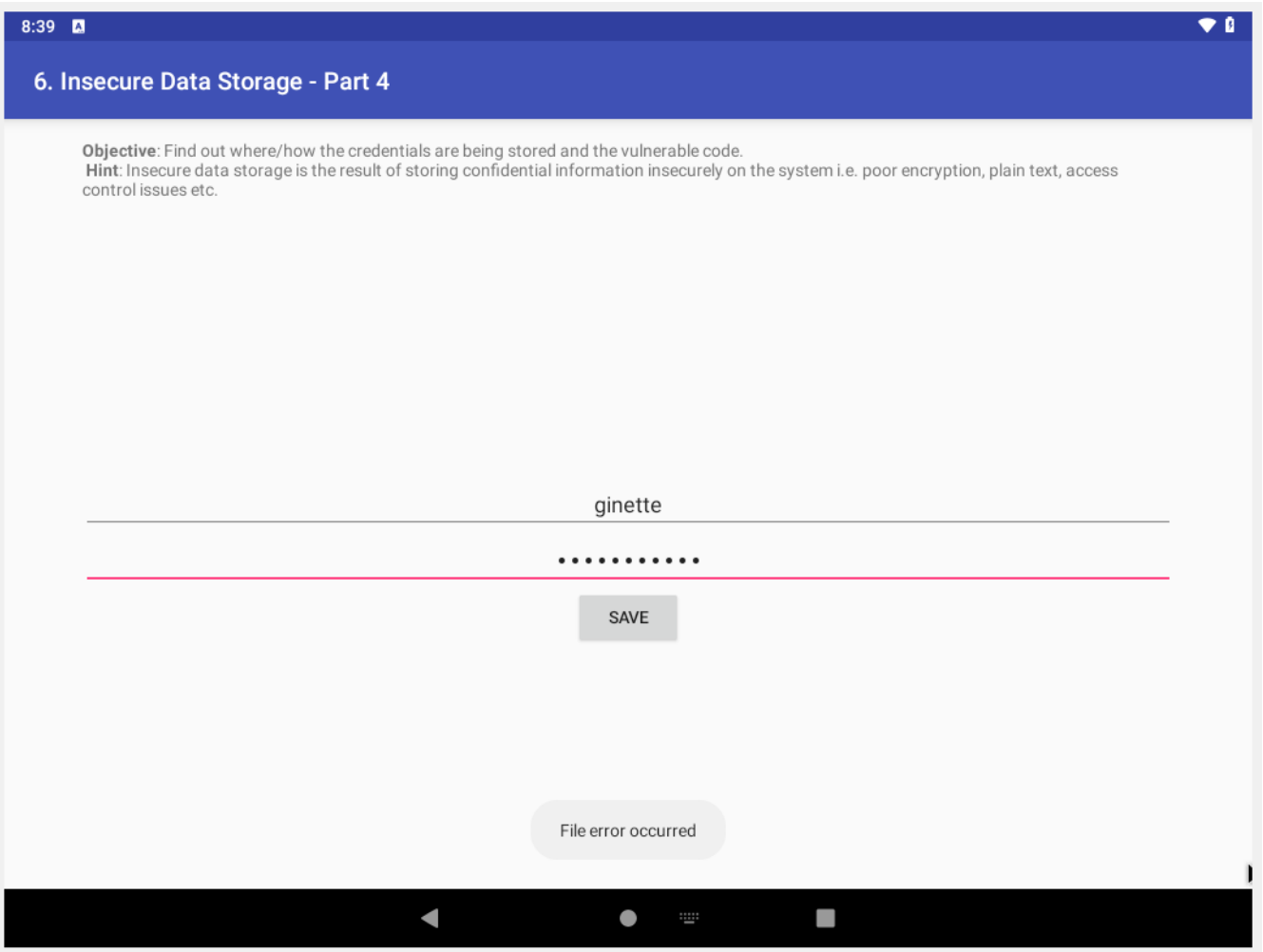
#5 insecure data storage part 3

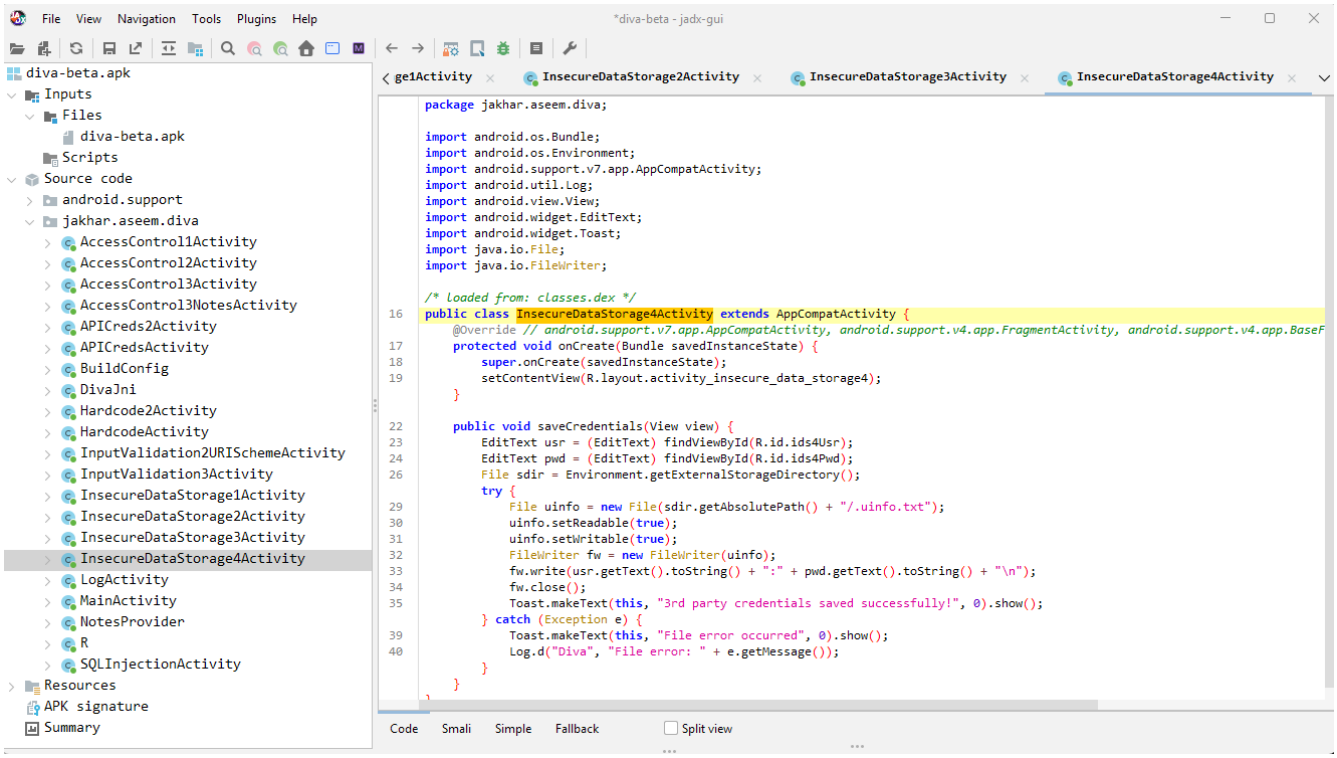

8:36
5. Insecure Data Storage - Part 3
Objective: Find out where/how the credentials are being stored and the vulnerable code.
Hint: Insecure data storage is the result of storing confidential information insecurely on the system i.e. poor encryption, plain text, access control issues etc.
pierrepaul
SAVE

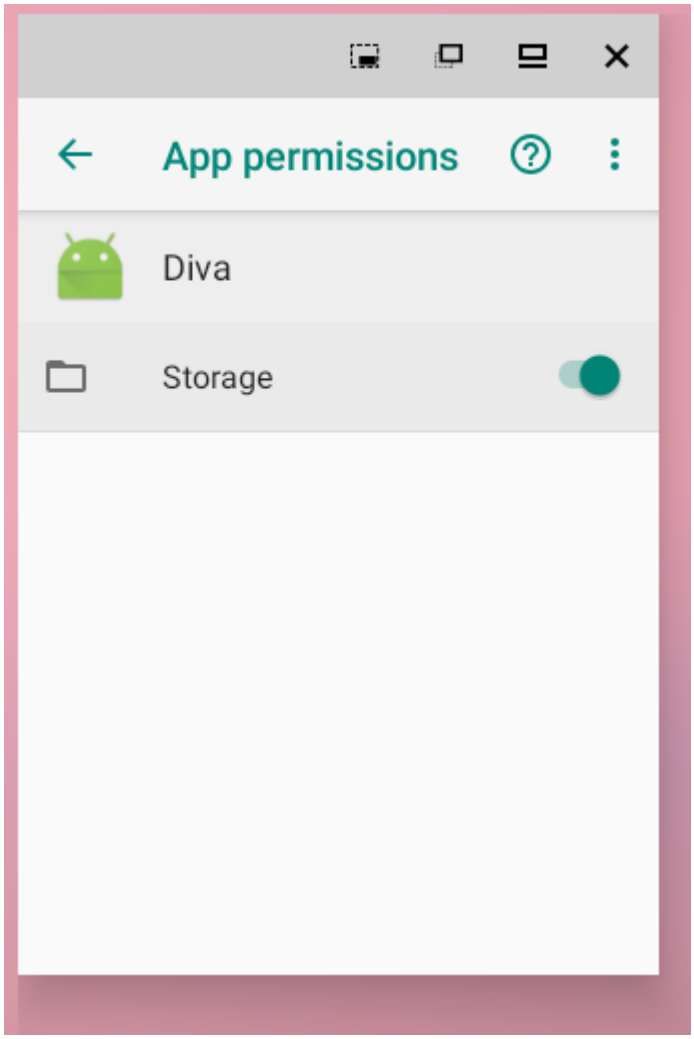


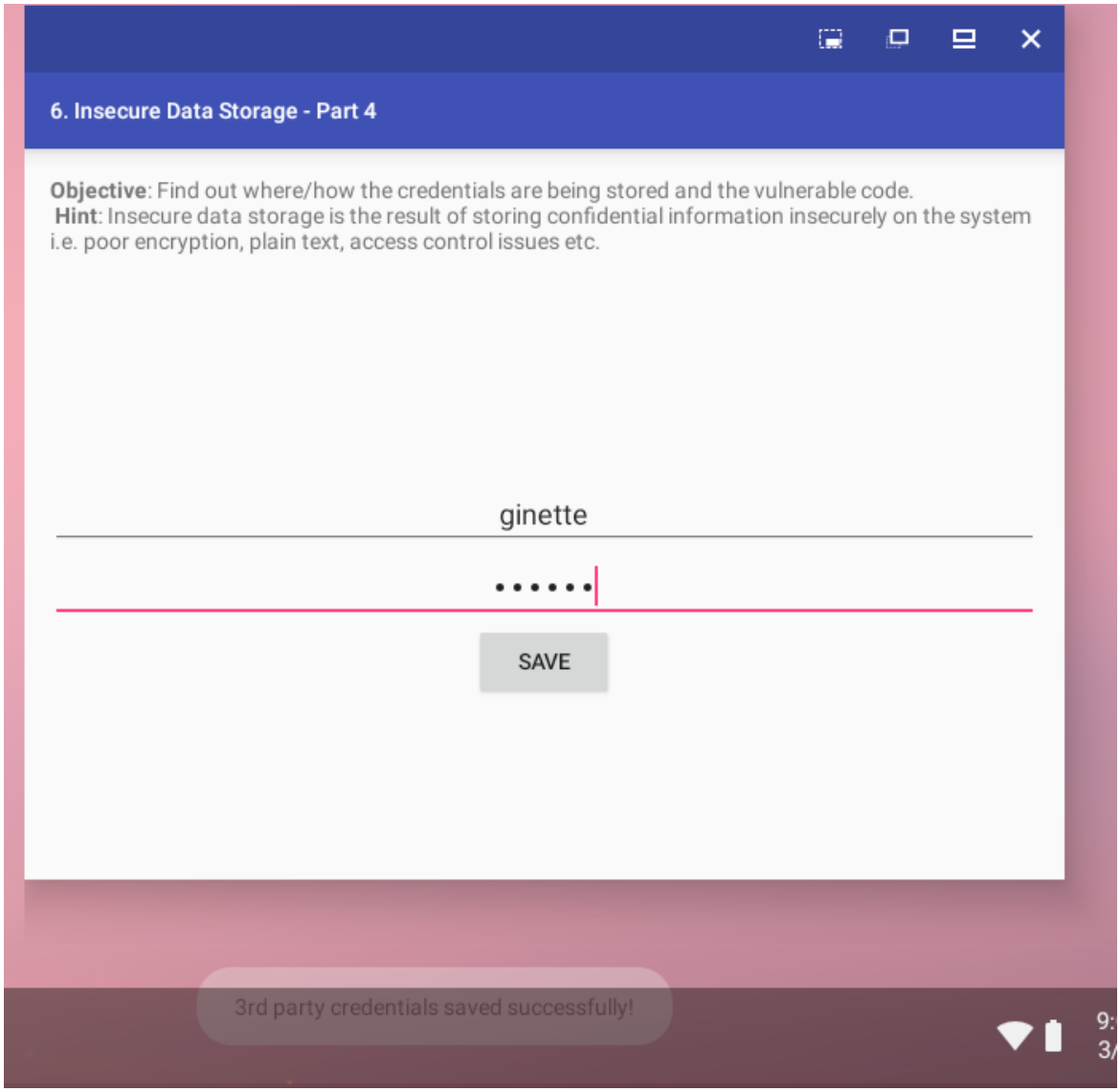

```
[3] + Stopped                 \vi ids2
console:/data/data/jakhar.aseem.diva/databases # cd ../
console:/data/data/jakhar.aseem.diva # cd databases/
console:/data/data/jakhar.aseem.diva/databases # ls
divanotes.db divanotes.db-shm divanotes.db-wal ids2
console:/data/data/jakhar.aseem.diva/databases # cd ../
console:/data/data/jakhar.aseem.diva # ls
cache code_cache databases shared_prefs uinfo11634577348591701 0tmp
console:/data/data/jakhar.aseem.diva # cat
cache/                        databases/                    uinfo11634577348591701 0tmp
code_cache/                   shared_prefs/
console:/data/data/jakhar.aseem.diva # cat
cache/                        databases/                    uinfo11634577348591701 0tmp
code_cache/                   shared_prefs/
console:/data/data/jakhar.aseem.diva # cat uinfo11634577348591701 0tmp
pierrepaul:lamirande
console:/data/data/jakhar.aseem.diva #
```

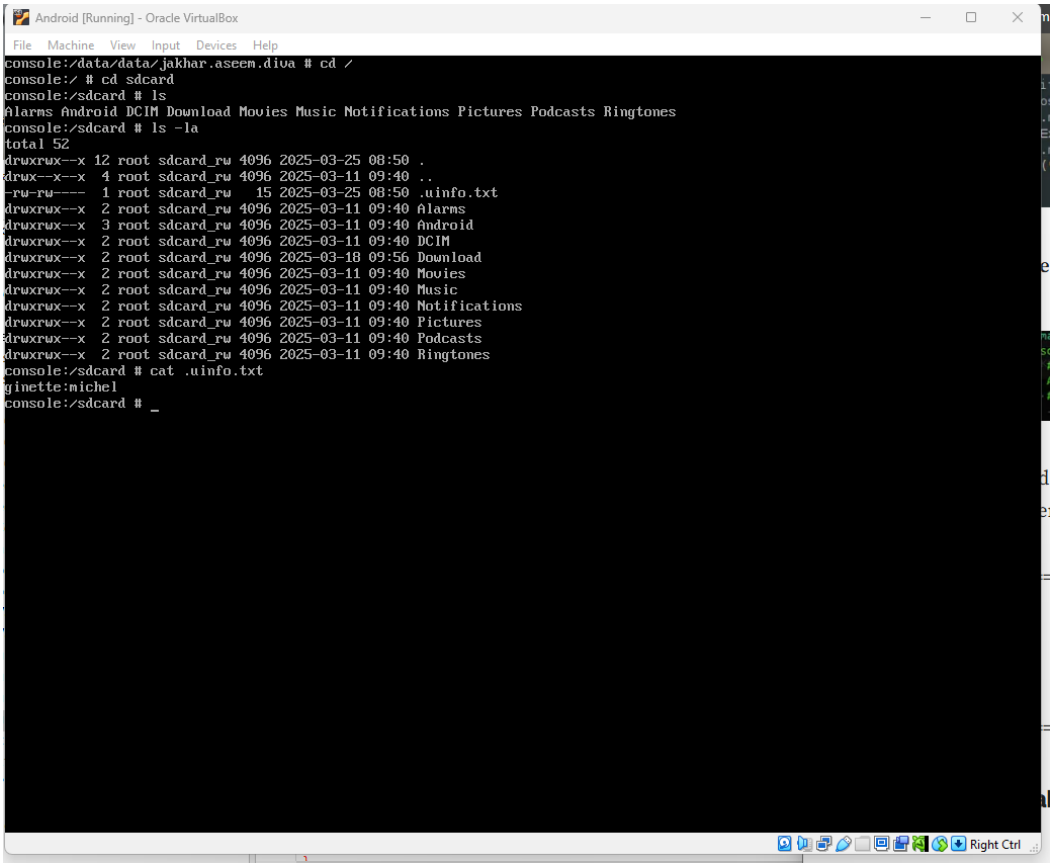
#6 insecure data storage


8:39
6. Insecure Data Storage - Part 4
Objective: Find out where/how the credentials are being stored and the vulnerable code.
Hint: Insecure data storage is the result of storing confidential information insecurely on the system i.e. poor encryption, plain text, access control issues etc.
ginette
SAVE
File error occurred

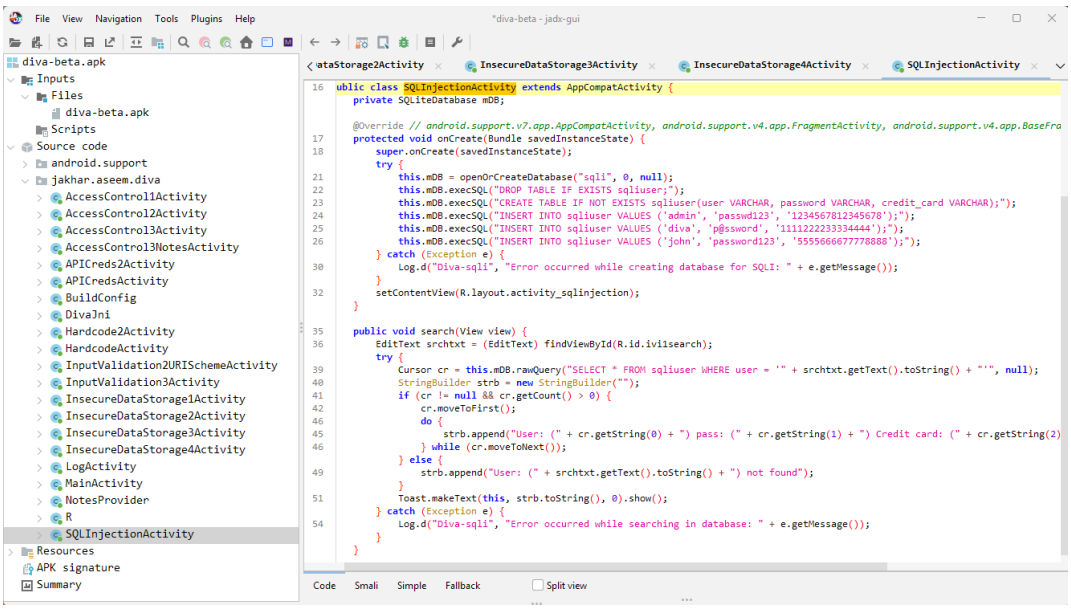

File View Navigation Tools Plugins Help
*diva-beta - jadx-gui
diva-beta.apk
Inputs
Files
diva-beta.apk
Scripts
Source code
android.support
jakhar.aseem.diva
AccessControl1Activity
AccessControl2Activity
AccessControl3Activity
AccessControl3NotesActivity
APICreds2Activity
APICredsActivity
BuildConfig
DivaJni
Hardcode2Activity
HardcodeActivity
InputValidation2URISchemeActivity
InputValidation3Activity
InsecureDataStorage1Activity
InsecureDataStorage2Activity
InsecureDataStorage3Activity
InsecureDataStorage4Activity
LogActivity
MainActivity
NotesProvider
R
SQLInjectionActivity
Resources
APK signature
Summary
ge1Activity
InsecureDataStorage2Activity
InsecureDataStorage3Activity
InsecureDataStorage4Activity
package jakhar.aseem.diva;
import android.os.Bundle;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import java.io.File;
import java.io.FileWriter;
/* loaded from: classes.dex */
public class InsecureDataStorage4Activity extends AppCompatActivity {
@Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.BaseF
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_insecure_data_storage4);
}
public void saveCredentials(View view) {
EditText usr = (EditText) findViewById(R.id.ids4Usr);
EditText pwd = (EditText) findViewById(R.id.ids4Pwd);
File sdir = Environment.getExternalStorageDirectory();
try {
File uinfo = new File(sdir.getAbsolutePath() + "/.uinfo.txt");
uinfo.setReadable(true);
uinfo.setWritable(true);
FileWriter fw = new FileWriter(uinfo);
fw.write(usr.getText().toString() + ":" + pwd.getText().toString() + "\n");
fw.close();
Toast.makeText(this, "3rd party credentials saved successfully!", 0).show();
} catch (Exception e) {
Toast.makeText(this, "File error occurred", 0).show();
Log.d("Diva", "File error: " + e.getMessage());
}
}
}
Code
Smali
Simple
Fallback
Split view

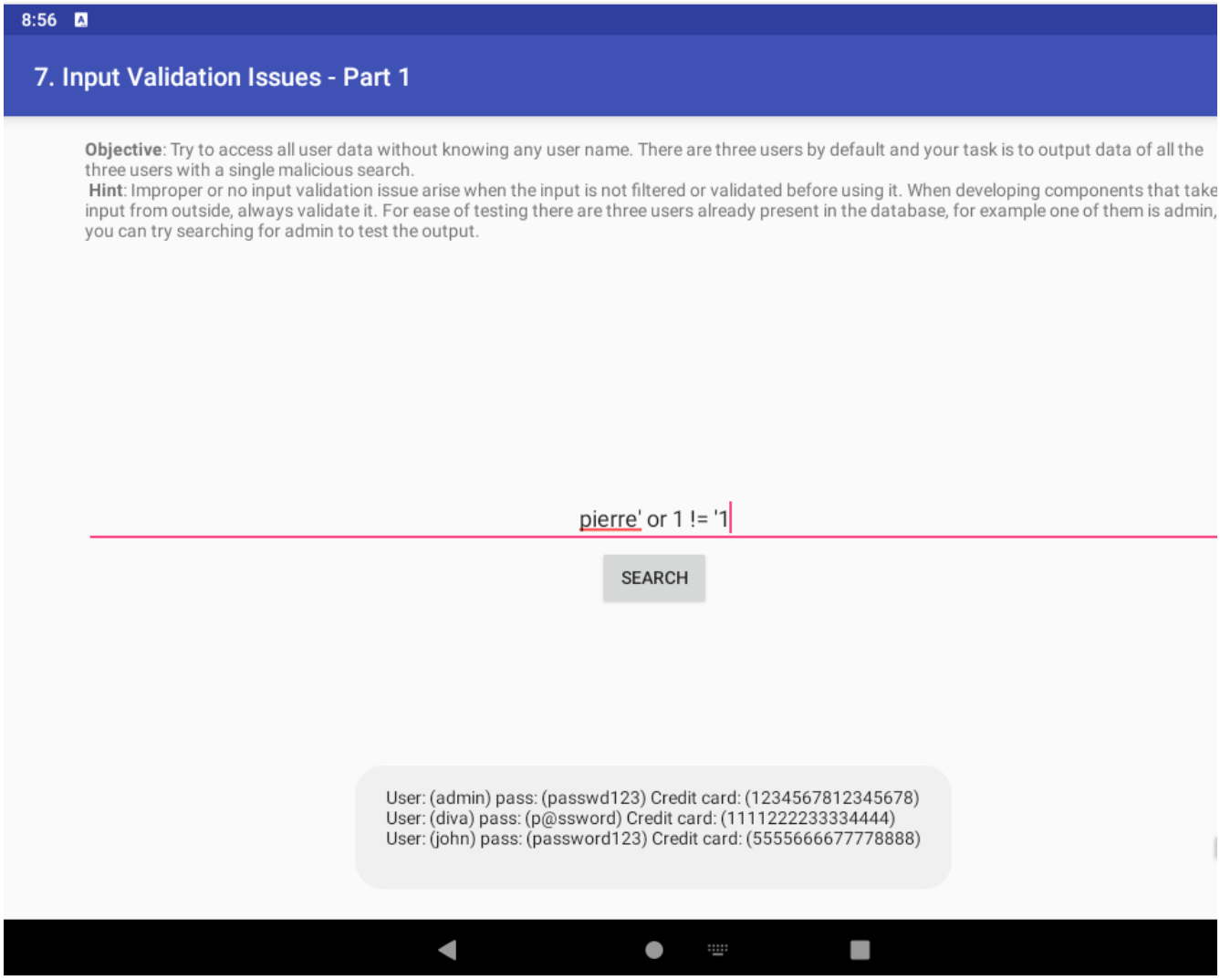

App permissions
Diva
Storage


6. Insecure Data Storage - Part 4
Objective: Find out where/how the credentials are being stored and the vulnerable code.
Hint: Insecure data storage is the result of storing confidential information insecurely on the system i.e. poor encryption, plain text, access control issues etc.
ginette
SAVE
3rd party credentials saved successfully!



#7 input validation issues part 1





fin du laboratoire.